IN THE SPECIFICATION:

Please amend the specification as follows:

From page 13, line 14 through page 14, line 4 of the original filed specification, delete the

following paragraph:

"Again, it bears repeating that the invention is not limited to trigonometric,

hyperbolic, and square root functions. The scope of the invention includes any family of

related functions. Note that the knowledge of these families of related functions need not be

encoded in the compiler itself. It may be preferred that the definitions for function families

are contained in a separate look-up table or other data store. For example, data store may

include ascii files, binary data files, database files, and more. The benefit of this

implementation is that defining new families does not require changes to the actual compiler

executable; the compiler may continue to work in the same manner regardless of the

information in the data store. Another benefit from such an implementation is that it may be

possible for to add custom families to the data store and receive the same efficiency

improvements from custom defined families as from common sets of functions like the

trigonometric or hyperbolic functions discussed above. Also, it is seen that no special

knowledge is required on the part of the programmer. The programmer writes code in a

standard language (C, C++, J++, Fortran, etc). Thus, portability of the source code is

maintained."

and add the following paragraph in its place:

Again, it bears repeating that the invention is not limited to trigonometric, hyperbolic, and square root functions. The scope of the invention includes any family of related functions. Note that the knowledge of these families of related functions need not be encoded in the compiler itself. It may be preferred that the definitions for function families are contained in a separate look-up table or other data store. For example, data store may include ascii files, binary data files, database files, and more. The benefit of this implementation is that defining new families does not require changes to the actual compiler executable; the compiler may continue to work in the same manner regardless of the information in the data store. Another benefit from such an implementation is that it may be possible [[for]] to add custom families to the data store and receive the same efficiency improvements from custom defined families as from common sets of functions like the trigonometric or hyperbolic functions discussed above. Also, it is seen that no special knowledge is required on the part of the programmer. The programmer writes code in a standard language (C, C++, J++, Fortran, etc). Thus, portability of the source code is maintained.